Carderock Division, Naval Surface Warfare Center

West Bethesda, Maryland 20817-5700

CRDKNSWC/HD-0011-26 December 1997

Hydromechanics Directorate
Research and Development Report

INTEGRATION OF A VISCOUS FLOW RANS SOLVER WITH AN UNSTEADY PROPULSOR FORCE CODE

BY

IRENEUSZ ZAWADZKI
DONALD FUHS
AND
JOSEPH GORSKI

Approved for public release; distribution is unlimited.

CRDKNSWC/HD-0011-26 INTEGRATION OF A VISCOUS FLOW RANS SOLVER WITH AN UNSTEADY PROPULSOR FORCE CODE

MAJOR CARDEROCK DIVISION TECHNICAL COMPONENTS

CODE	011	Director of Technology
	10	Machinery Systems/Programs and Logistics Directorate
	20	Ship Systems and Programs Directorate
	50	Hydromechanics Directorate
	60	Survivability, Structures and Materials Directorate
	70	Signatures Directorate
	80	Machinery Research and Development Directorate
	90	Machinery In-Service Engineering Directorate

CARDEROCK DIVISION, NSWC, ISSUES THREE TYPES OF REPORTS:

1. **CARDEROCKDIV reports, a formal series,** contain information of permanent technical value. They carry a consecutive numerical identification regardless of their classification or the originating directorate.

2. **Directorate reports, a semiformal series,** contain information of a preliminary, temporary, or proprietary nature or of limited interest or significance. They carry an alphanumerical identification issued by the originating directorate.

3. **Technical memoranda, an informal series,** contain technical documentation of limited use and interest. They are primarily working papers intended for internal use. They carry an identifying number which indicates their type and the numerical code of the originating directorate. Any distribution outside CARDEROCKDIV must be approved by the head of the originating directorate on a case-by-case basis.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1997

3. REPORT TYPE AND DATES COVERED
Research and Development, Oct 96 to Sept 97

4. TITLE AND SUBTITLE
Integration of a Viscous Flow RANS Solver with an Unsteady Propulsor Force Code

5. FUNDING NUMBERS

6. AUTHOR(S)

Ireneusz Zawadzki, Donald Fuhs and Joseph Gorski

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Hydromechanics Directorate, Code 5400
Carderock Division
Naval Surface Warfare Center
9500 MacArthur Boulevard
West Bethesda, MD 20817-5700

8. PERFORMING ORGANIZATION REPORT NUMBER
CRDKNSWC/HD-0011-26

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Ship Structures and Systems
S&T Division 334
800 N. Quincy Street
Arlington, VA 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The unsteady propeller force code PUF-2.1 has been coupled with DTNS - a viscous RANS flow solver. The effects of the propeller are modeled via a spatially varying time averaged body force distribution incorporated in the RANS equations. Simple test cases are presented showing that the code properly handles an open water flow as well as a circumferentially varying inflow. Differences between the new code and its steady counterpart are highlighted, both from the standpoint of the code implementation as well as the calculated physical results.

The work is easily extendable to the parallel version of UNCLE and maneuvering calculations performed as part of the DOD Challenge Program when the unsteady forces and propulsor effect are obtained from PUF as a precursor to the full RANS calculations of the entire boat with propulsor.

14. SUBJECT TERMS
Propellers, unsteady forces, Reynolds Averaged Navier Stokes RANS, body forces, PUF-2.1, UNCLE, DOD Challenge Program

15. NUMBER OF PAGES
19

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT

CONTENTS

FIGURES

TABLE

NOTATION

D	Propeller diameter
$\hat{D}$	Non-dimensional propeller diameter
$f_{x,y,z}$	RANS body force density components
$\hat{f}_{x,y,z}$	Non-dimensional RANS body force density
$\delta\hat{F}^{i}_{x,y,z}$	Cartesian components of non-dimensional body force acting on the ith propeller panel
J	Advance ratio
K_T	Thrust coefficient
K_Q	Torque coefficient
L_0	Length scale used for non-dimensionalizing the RANS equations
n	Shaft speed, revolutions per second
N_B	Number of blades
P^i	Proportion of the grid cell which intersects the ith propeller panel
V	Ship speed
$\hat{V}p^i$	Non-dimensional volume swept by the propeller panel
Q	Torque
T	Thrust
$W_{i,j,k}$	Linear interpolation weight coefficient
x,y,z	Cartesian coordinates
θ	Circumferential angle
ξ, η, ζ	Parametric coordinates

ABSTRACT

The unsteady propeller force code PUF-2.1 has been coupled with DTNS - a viscous RANS flow solver. The effects of the propeller are modeled via a spatially varying time averaged body force distribution incorporated in the RANS equations. Simple test cases are presented showing that the code properly handles an open water flow as well as a circumferentially varying inflow. Differences between the new code and its steady counterpart are highlighted, both from the standpoint of the code implementation as well as the calculated physical results.

The work is easily extendable to the parallel version of UNCLE and maneuvering calculations performed as part of the DOD Challenge Program when the unsteady forces and propulsor effect are obtained from PUF as a precursor to the full RANS calculations of the entire boat with propulsor.

ADMINISTRATIVE INFORMATION

The work described in this report was performed by the Signatures Control Technology Department of the Signatures Directorate and by the Propulsor Department of the Hydromechanics Directorate, Carderock Division, Naval Surface Warfare Center. The work was sponsored by the Office of Naval Research, Ship Structures and Systems S&T Division (Code 334), under the Advanced Propulsor Task of the FY97 Submarine Hull, Mechanical, and Electrical Technology Program (PE0602121N).

OBJECTIVES

The long term objective of this project is to develop an automated procedure for including the effect of a propulsor in RANS calculations of a ship or submarine hull flow. The goal for FY97 work was to develop a working version of the coupling procedure using available proven versions of RANS and unsteady propeller force codes.

BACKGROUND

A method for incorporating propeller unsteady forces in hull flow calculations was developed by Stern et. al. [1]. The method relies on an iterative procedure in which the RANS calculated flow becomes the input to the unsteady propeller performance code and, in turn, the calculated propeller forces are translated into a body force term in the RANS equations. The body forces are allowed to vary circumferentially and radially to simulate the angle dependent forces acting on the rotating propeller blade elements. This representation allows inclusion of the important propeller effects into the RANS flow solution without the complication of gridding moving blade surfaces.

Tseng, Nguyen and Fuhs [2] started an effort of implementing the body force representation to couple the David Taylor Navier Stokes RANS code (DTNS) [3] with MIT's Propeller Unsteady Performance (PUF) program [4]. They compared propeller force and torque coefficient predictions using two versions of the MIT PUF code: PUF-2.1 and PUF-3A. They also modified both codes to extract time averaged body forces acting on individual propeller grid panels. The present work is a continuation of their project.

RANS/PUF COUPLING PROGRAM

A steady version of the RANS/propeller interaction analysis system has been developed by Weems & Korpus [5] as part of a project funded by ARPA Advanced Submarine Technology Program. Their code allows, as one of the options, coupling DTNS with MIT's Propeller Steady Force (PSF) code. It was therefore a convenient starting point for the present work. The basic structure of the code is directly applicable to the unsteady case. The words "steady" and "unsteady" refer here to the propeller force code used and not to the RANS flow solution which in both cases is time averaged. A steady propeller code such as PSF produces an axisymmetric body force, whereas an unsteady propeller code such as PUF-2.1 calculates a non-axisymmetric body force in response to the non-axisymmetric hull stern flow.

Major changes had to be made to allow multiple circumferential harmonics of the inflow and propeller loading. Some additional rewriting of the code was done to simplify the inflow velocity interpolation procedure and to speed up the setup stage of the code. Since reference [5] gives a detailed description of the steady (RANS-PSF) coupling program and since our code follows their methodology, in general only a brief outline of the unsteady version (RANS-PUF) will be given in this report. More detailed discussion is provided occasionally to elucidate several important differences between the two codes.

SETTING UP THE RANS/PUF ITERATIONS

As in the steady version, our program is divided into two separate FORTRAN modules: RANS-PUF-SETUP and RANS-PUF-ITER. Figures 1 and 2 show flow diagrams for both modules. RANS-PUF-SETUP is run once at the beginning of the calculations. Its role is to compute and save interpolation coefficients that are used during the iteration stage to relate flow and force information between the RANS and PUF codes. The code is divided into 6 steps shown in the diagram in Fig. 1. After reading the RANS grid file and the propeller definition file (steps 1 and 2) the program issues a command to run the propeller boundary value problem code: PUF2BVP. (The

PUF modules are called using a C language utility "system()" which is linked with the main FORTRAN program. The same technique is used in [5]). Propeller panel geometry is then read (step 3) and the volume swept by each panel is calculated (step 4). In step 5, the program performs a search of the RANS grid to identify grid cells coinciding with the volume swept by the propeller. For each such cell, its volume, the circumferential location of its center and the cell/panel overlap fraction is calculated and saved to a file. Finally, in step 6, another search is performed to find grid cells intersecting the plane where the propeller inflow is to be evaluated. The steps described above mirror those used in [5] for the steady case.

Several procedures had to be, of course, modified. First, a relatively minor change is using the PUF definition file as input for the propeller geometry, rather than preparing a separate file of a different format and having the SETUP program reconstruct the PUF input file. This simplifies the task of preparing the input data especially for someone already familiar with running the PUF code. The procedure for interpolating body forces onto the RANS grid in the steady code [5] was entirely adaptable to the unsteady case. In order to redistribute circumferentially varying body forces onto the grid, the circumferential coordinate, (angle θ), of each swept grid cell must be known. Fortunately, the steady code computes and saves this information. To assure finding a complete list of body force cells, a three-pass search procedure is used that checks for all possible intersections of grid cell faces and edges with the rotating edges of the propeller panels (see [5] for details). To improve the computational efficiency of the search we have added an additional sorting subroutine in step 5 of the unsteady code. The new subroutine rapidly filters out grid cell - propeller panel pairs that are sufficiently apart that they cannot possibly intersect. As a result of this improvement the CPU time required to run the "SETUP" stage of the code was reduced approximately by a factor of 10.

The final step, 6, is the calculation of interpolation weights for mapping RANS inflow velocity onto a regular planar mesh, which is used to compute inflow velocity harmonics. In the steady version, circumferential averaging is applied to filter out the non-zero harmonics in the RANS computed flow. In order to perform the averaging, an additional list of RANS grid cells swept by each propeller panel had to be generated. In the unsteady code, the averaging is not needed. Therefore, we decided to completely rewrite the inflow interpolation procedure rather than trying to reuse the steady code approach, which would be unnecessarily complicated for our purposes. Our new flow interpolation procedure works as follows: for a given point P on the inflow plane, a search of the RANS grid is performed to find the cell which contains P. Tri-linear interpolation is used to compute the field quantity at point P. In other words, if we label the cell's corners with indices (i, j, k) where i, j, k are either 0 or 1, then the field quantity value at point P, F_p, is given by:

$$F_P = \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} W_{i,j,k} F_{i,j,k} \qquad (1)$$

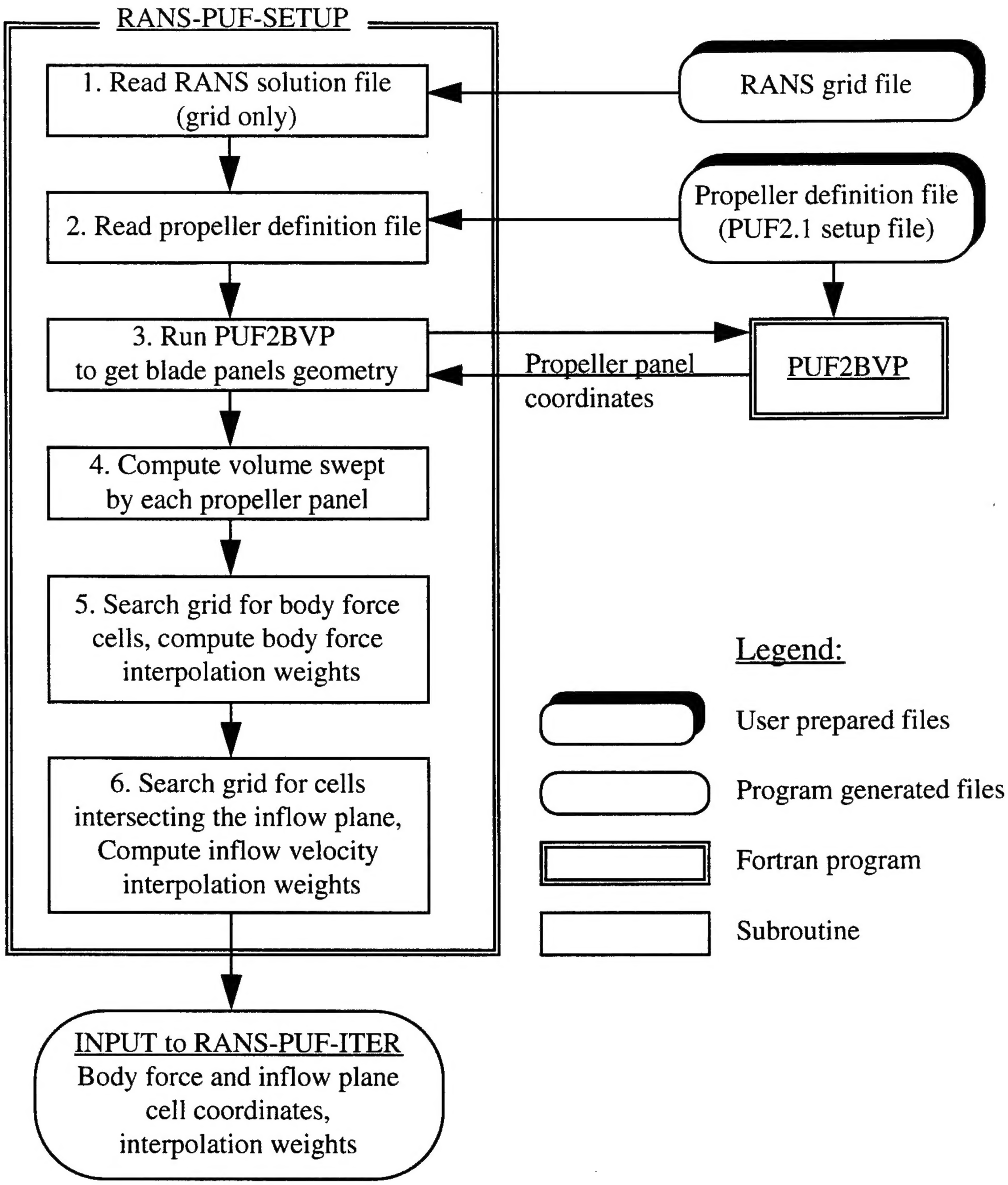
RANS-PUF-SETUP
1. Read RANS solution file (grid only)
2. Read propeller definition file
3. Run PUF2BVP to get blade panels geometry
4. Compute volume swept by each propeller panel
5. Search grid for body force cells, compute body force interpolation weights
6. Search grid for cells intersecting the inflow plane, Compute inflow velocity interpolation weights
RANS grid file
Propeller definition file (PUF2.1 setup file)
PUF2BVP
Propeller panel coordinates
INPUT to RANS-PUF-ITER Body force and inflow plane cell coordinates, interpolation weights
Legend:
User prepared files
Program generated files
Fortran program
Subroutine

Fig. 1. RANS-PUF-SETUP flow diagram.

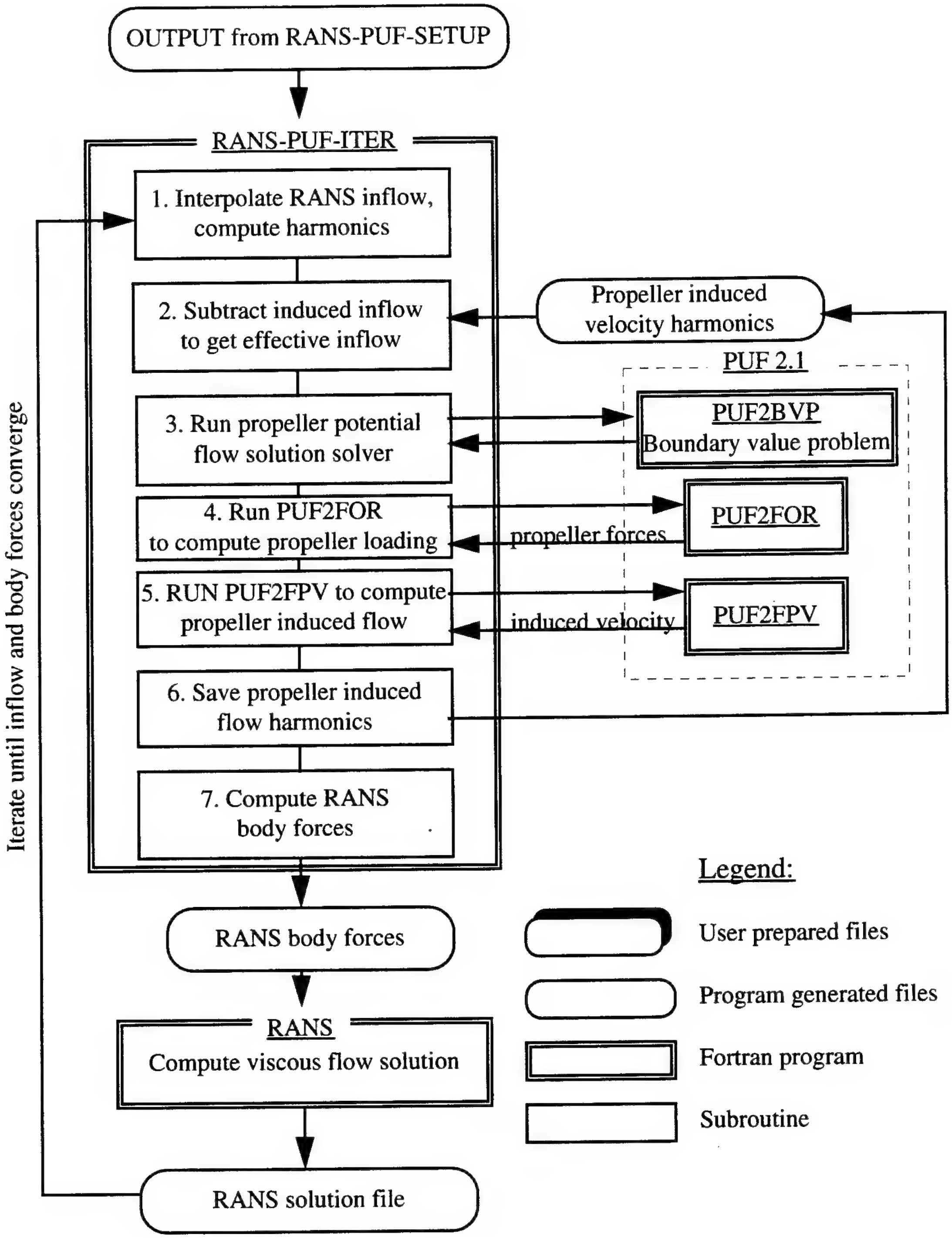
OUTPUT from RANS-PUF-SETUP
RANS-PUF-ITER
1. Interpolate RANS inflow, compute harmonics
2. Subtract induced inflow to get effective inflow
3. Run propeller potential flow solution solver
4. Run PUF2FOR to compute propeller loading
5. RUN PUF2FPV to compute propeller induced flow
6. Save propeller induced flow harmonics
7. Compute RANS body forces
Propeller induced velocity harmonics
PUF 2.1
PUF2BVP
Boundary value problem
PUF2FOR
propeller forces
PUF2FPV
induced velocity
Iterate until inflow and body forces converge
RANS body forces
RANS
Compute viscous flow solution
RANS solution file
Legend:
User prepared files
Program generated files
Fortran program
Subroutine

Fig. 2. RANS-PUF-ITER flow diagram.

To define the weights, $W_{i,j,k}$, we introduce a local parameterization of the grid coordinates: (x, y, z) -> (ξ, η, ζ) with $0<\xi, \eta, \zeta<1$ which maps the cell's interior into a unit volume in the parametric space. Let, ξ_P, η_P, ζ_P be the coordinates of P in the parametric space. Then:

$$W_{i,j,k} = |1-\xi_P| \cdot |1-\eta_P| \cdot |1-\zeta_P| \qquad (2)$$

In other words, the weights $W_{i,j,k}$ are equal to the volumes into which the cell is divided by three planes: $\xi = \xi_P$, $\eta = \eta_P$ and $\zeta = \zeta_P$ in the parametric space. Figure 3 provides a two-dimensional graphical illustration of the interpolation procedure (the volume weights become areas in two dimensions). In practice, finding the parametric coordinates of point P is accomplished by a three-dimensional extension of the half-interval method [6]. Namely, the cell is divided into eight smaller sub-cells, then the sub-cell that contains P is selected and subdivided again. This subdivision and selection process is repeated until the remaining sub-cell's size is less than the desired accuracy. For example, 6 iterations allows pinpointing ξ_P, η_P, ζ_P with better than 1% precision. Summarizing, the output of step 6 is a list containing DTNS block number, cell indices and values of ξ_P, η_P, ζ_P for all grid points on the inflow plane.

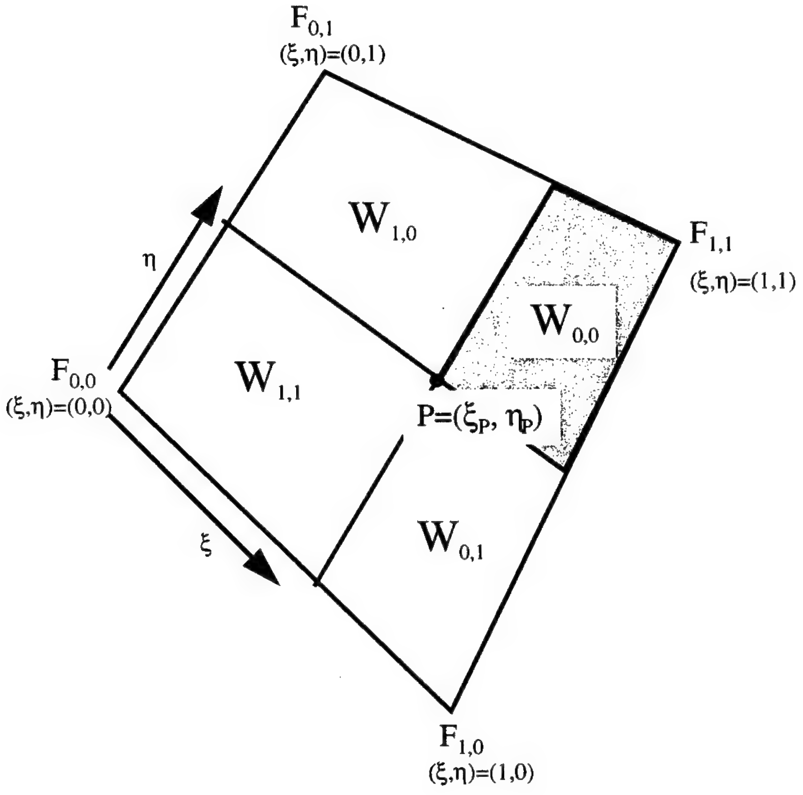

Fig. 3. Two-dimensional illustration of the linear interpolation procedure.

RANS-PUF ITERATION PROCEDURE

The second program, called RANS-PUF-ITER, serves as "coordinator" of the RANS / PUF iteration process. Its primary role is to translate the unsteady propeller forces into a distribution of body forces to be used by the RANS flow solver. Figure 2 shows the flow diagram for this part of the program. The first step is to read the RANS calculated flow from the previous iteration (or some initial "guess" if this is the first iteration) and find the values of the velocity components on a user defined inflow plane located 1/4 to 1/2 of a propeller radius upstream of the propeller. Typically, the velocity is determined at 60 equally spaced circumferential locations for several (typically 10 to 15) radii. The program uses here the interpolation coefficients calculated in step 6 of RANS-PUF-SETUP (recall Eq. (1) and (2)). Interpolated values at each radial location are Fourier transformed to obtain a set of inflow harmonics. The propeller induced flow harmonics, saved from the previous iteration, are subtracted to obtain the effective inflow harmonics (step 2). The number of circumferential locations on the inflow plane grid should be selected to resolve the local flow gradients. A number of interpolation points larger than 60 may be more appropriate if, for example, the inflow to the propeller contains a very thin wake or a localized vortex.

Next, the program calls the PUF boundary value problem solver: PUF2BVP (step 3) followed by a call to PUF2FOR (step 4) which computes the propeller forces. The time averaged forces are computed and saved for each propeller panel at several evenly distributed circumferential locations (typically 60). Once again, the number of circumferential points should be selected to obtain a sufficiently smooth representation of the angular dependence of the panel forces. In step 5 another PUF module (PUF2FPV) is called to evaluate the propeller induced velocity field just upstream of the propeller. The induced velocity vectors are evaluated at the same user defined uniform plane mesh as the one used in step 1. Axial, tangential and radial components are Fourier transformed and the harmonics saved in a file to be used in step 2 during the next program iteration.

The incremental propeller force values computed in step 4 can be interpolated to obtain force acting on each panel for an arbitrary value of circumferential angle, θ. The propeller coordinate system defining the angle θ is shown in Figure 4.

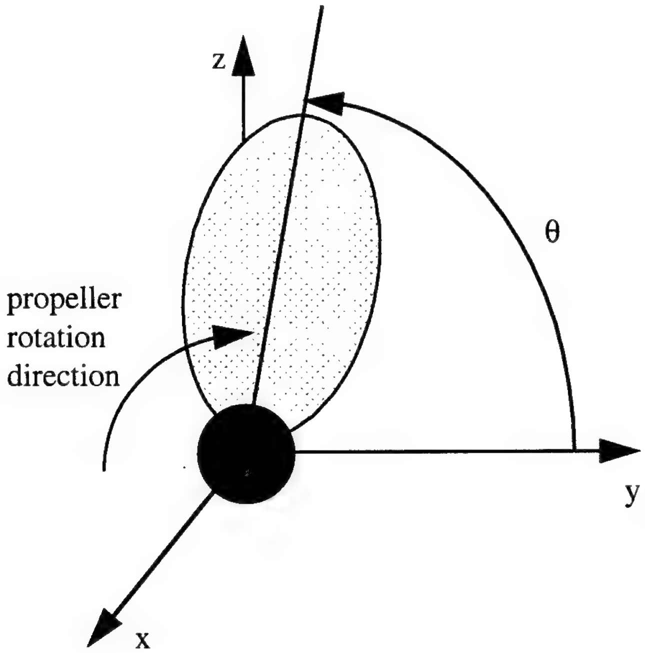

Fig. 4. Propeller coordinate system.

At this point we have all the ingredients required to compute the Cartesian components of the RANS body force density f_x, f_y, f_z (step 7). The body force must be evaluated at the centers of all grid cells that intersect the volume swept by the propeller. A list of those cells is compiled and saved in step 5 of the SETUP program. The body force density is defined by the following equations:

$$\hat{f}_{x,y,z} = \frac{\hat{D}^2}{J^2} N_B \sum_{i=1}^{N_p} \frac{P^i\, \delta\hat{F}^i_{x,y,z}(\theta)}{\hat{V}p^i} \tag{3}$$

where $\hat{D}$ is the propeller diameter, J is the advance ratio, N_B is the number of blades, P^i is the proportion of the grid cell which intersects the ith panel, $\delta\hat{F}^i_{x,y,z}(\theta)$ are the Cartesian components of force acting on that panel, θ is the circumferential coordinate of the grid cell center and $\hat{V}p^i$ is the volume swept out by the propeller panel. The summation is over all propeller panels that sweep through the given cell. The caret symbol ^ in Eq. (3) indicates that the quantity is non-dimensional. The non-dimensionalization is defined as follows:

$$\hat{f}_{x,y,z} = \frac{\text{Body force per unit volume}}{\rho U^2} L_0 \tag{3a}$$

$$\delta\hat{F}^{i}_{x,y,z} = \frac{\text{Force acting on panel}}{\rho n^{2} D^{4}} \tag{3b}$$

$$\hat{D} = \frac{D}{L_0}, \qquad \hat{V}p^{i} = \frac{Vp^{i}}{L_0^{3}} \tag{3c}$$

where L_0 is a length scale chosen to non-dimensionalize the RANS equations. We should point out that Eq. (3) is almost identical to Eq. (8) given in Ref. [5]. (We use here the same notation to make the similarity readily apparent). This is not surprising since the unsteady propeller code in this work (PUF-2) and the steady version (PSF) in [5], use identical non-dimensionalization of variables. The only difference is that here we use circumferentially varying values of panel forces whereas in Ref. [5] the forces were independent of the angle θ. Since the original PUF-2 output does not provide the incremental panel forces required for reconstructing the angular dependence, the PUF-2 code had to be modified. Changes made by Tseng et. al. [2] enable one to compute and save in a file the axial (x-direction) component of the propeller panel forces. We have extended these changes to the two remaining (y and z) force components. We have also added an additional modification to the PUF code that allows computing and writing out the angular coordinate of each panel relative to some fixed reference angle. These relative angles are necessary to determine correctly the functions $\delta F^{i}_{x,y,z}(\theta)$ in Eq. (3).

Results of step 7 are saved in a file in a form of a list of relevant RANS grid cell indices and the corresponding force density components. The file is used as one of the inputs to the RANS code. The RANS iterations are restarted using the flow solution from previous iterations as an initial condition. The newly computed body force is also imposed on the flow field. The RANS code has to be iterated until the flow converges to another steady state, reflecting the new body force distribution. The number RANS iterations necessary is highly dependent on the complexity of the body/propeller system. The resulting flow solution is then compared to the results from the previous run of the RANS solver. If the two solutions are identical to within a predefined margin of accuracy, the RANS/PUF iteration process is stopped. Otherwise, another run of RANS-PUF-ITER is performed and the process is repeated. A UNIX shell script has been written to automate the somewhat tedious task of copying various files to the proper directories, restarting the appropriate programs in the proper sequence and saving the intermediate results for later post-processing. Besides providing convenience, the script significantly shortens the turn-around time for the calculations. The reason for this is that the RANS calculation may take a significant amount of time so the user may not be around to immediately initiate the next step. The script, on the other hand, waits patiently and starts the next program in the sequence as soon as the preceding step is completed. To avoid wasting computer resources, the script checks for existence of appropriate output files before proceeding with another RANS run.

PRELIMINARY VALIDATION RESULTS

A complete validation of the coupling code requires running a body/propeller configuration, preferably one with an experimental measurement database available for comparison. However, we initially wanted to concentrate on development of the coupling algorithm instead of grid generation or other code issues. Therefore, we opted for running consistency checks using a few simple configurations to ensure that the coupling algorithm works as planned, and to perform calculations for more complex configurations in next year's effort.

Our first test is an open water calculation. We have run the code for two different propellers: a 4-bladed MARIN B-series (operating at J=0.47) identical to the one used by Weems & Korpus in testing the steady case (Ref. [5]) and a 5-bladed FF1052 propeller (at J=0.773) for which an extensive numerical evaluation was performed by Tseng et. al. [2]. The propeller was placed in a cylindrical flow domain represented by a relatively coarse grid. The number of grid points was 80 x 30 x 72 in the axial, radial and circumferential directions respectively. The propellers were represented by a 9 by 10 panel lattice. Three hundred DTNS iterations were performed after each RANS-PUF-ITER step. The propeller's thrust coefficient K_T remained almost constant throughout the calculations. This is to be expected since for the open water case the flow induced by the propeller should match exactly the change in flow velocity due to body forces resulting in the same, uniform effective inflow for each iteration. Table 1 shows the history of the thrust and torque coefficients for the FF1052 case. The integrated and properly non-dimensionalized values of the total RANS body force and torque were also included for comparison. Ideally, one would expect the integrated body force and torque to match exactly K_T and K_Q calculated by PUF. Due to interpolation errors, some differences are unavoidable. However, as table 1 demonstrates, the differences are well below 1%, a quite acceptable error considering the low order interpolation method used and the coarseness of the grid. K_T and K_Q were calculated for the same propeller by Tseng et al [2]. The values they obtained: K_T =0.145 and K_Q =0.0275 (for J=0.773) are very close to those shown in Table 1.

Iteration No.	K_T (PUF)	RANS body force	% difference	K_Q (PUF)	RANS body force torque	% difference
0	0.14460	0.14495	0.24	0.02746	0.02754	0.31
1	0.14495	0.14490	0.04	0.02749	0.02754	0.16
2	0.14491	0.14500	0.06	0.02749	0.02753	0.14
3	0.14501	0.14448	0.37	0.02751	0.02745	0.20
4	0.14449	0.14457	0.06	0.02743	0.02747	0.14
5	0.14458			0.02745		

Table 1. Histories of thrust and torque coefficients and body force magnitude and moment for FF1052 open water test.

The converged values for the open water thrust and torque coefficients for the MARIN B-series propeller were: K_T =0.162, K_Q =0.0202. These results also compare reasonably well with the design values for this propeller: K_T =0.156, K_Q =0.0200 quoted in Ref. [5].

Besides the integrated force values, we also looked at spatial distribution of various field values after the last RANS-PUF-ITER iteration to assure that the interpolation procedures preserve the axial symmetry of the problem. For example, Figure 5 shows body force density integrated in the axial direction. The axial force is represented by a contour plot, while arrows depict the direction and magnitude of the in-plane force components. Clearly, the symmetry of force distribution is well preserved throughout the calculation. As a consequence, the total velocity field also remains almost perfectly symmetric (see Figure 6). The strong tangential component of body force shown in Figure 5 causes the flow to swirl downstream of the propeller. For the outer radii, the inward radial body force causes slipstream contraction. There is no hub model in PUF-2.1, so PUF-2.1 allows strong radial flows and radial body forces for the inner radii that should not be present. PUF-14, the next generation of the propeller code currently under development, will include a hub model.

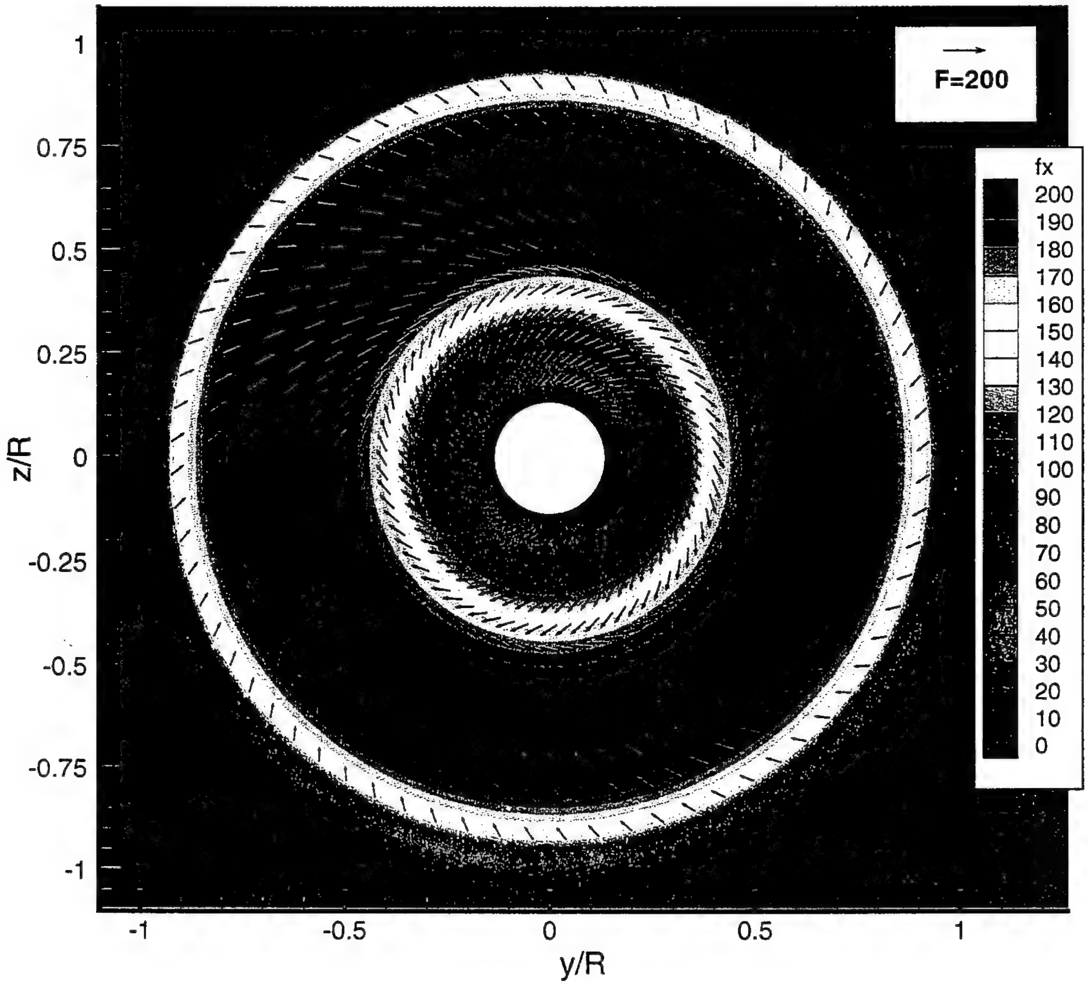

Fig. 5. Open water calculation, RANS body force distribution.

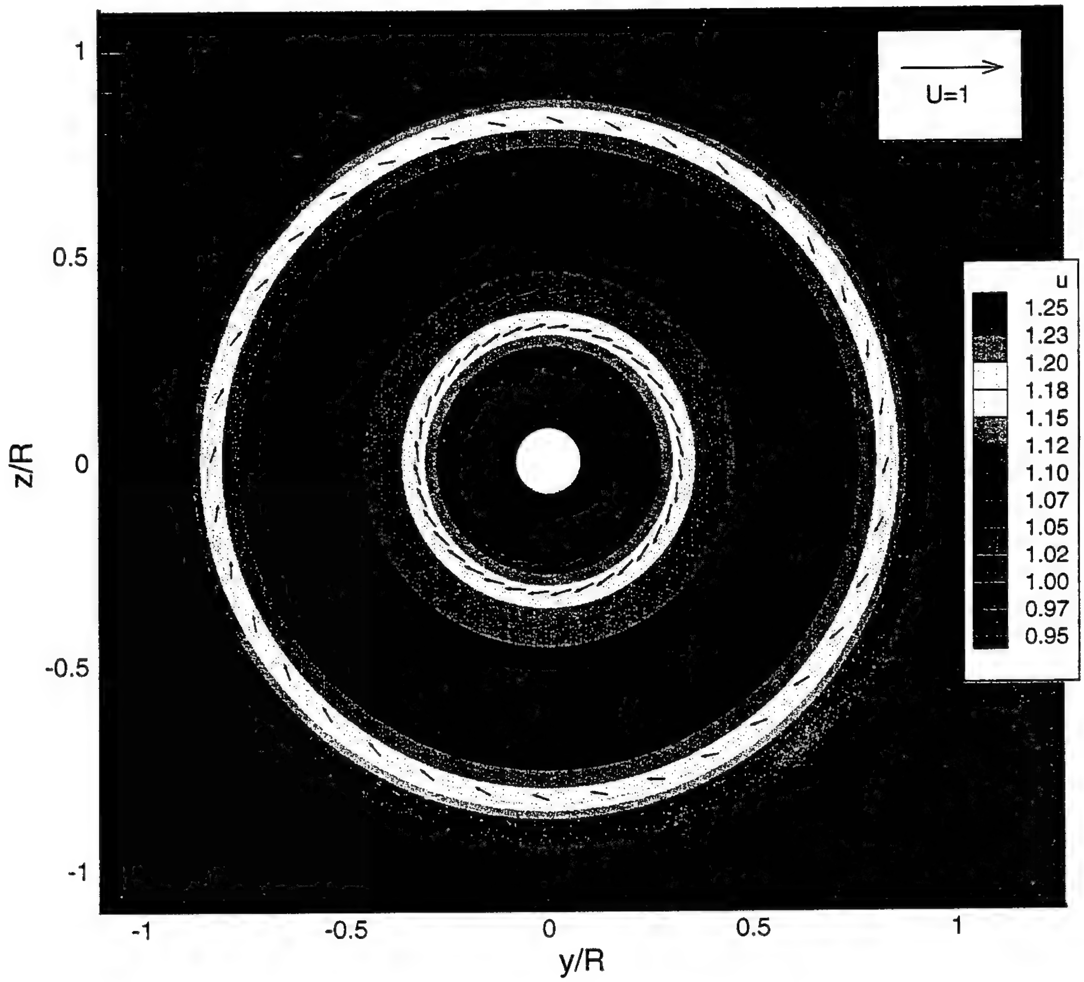

Fig. 6. Open water calculation, velocity field downstream of the propeller.

The open water test proved very useful as a "debugging" case for our code. However, it does not demonstrate the crucial capability of the RANS-PUF system to compute a circumferentially varying force distribution. In fact, the open water flow reduces the problem to a steady case that can be handled by the RANS-PSF system [5]. Therefore, in our second test case we have introduced an arbitrary "wake" into the inflow. Figure 7 shows the inflow velocity contours. The velocity deficit is described by a cosine function of the circumferential angle θ with the minimum at $\theta=80^{\circ}$ and transitioning smoothly to the free stream value at $\pm 45^{\circ}$ from the wake's center . There is no velocity variation imposed in the radial direction. Obviously, the "wake" was not meant to represent any practical application. All we needed for this test was an inflow that introduces non-zero velocity harmonics and yet is simple enough so that the results can be easily interpreted.

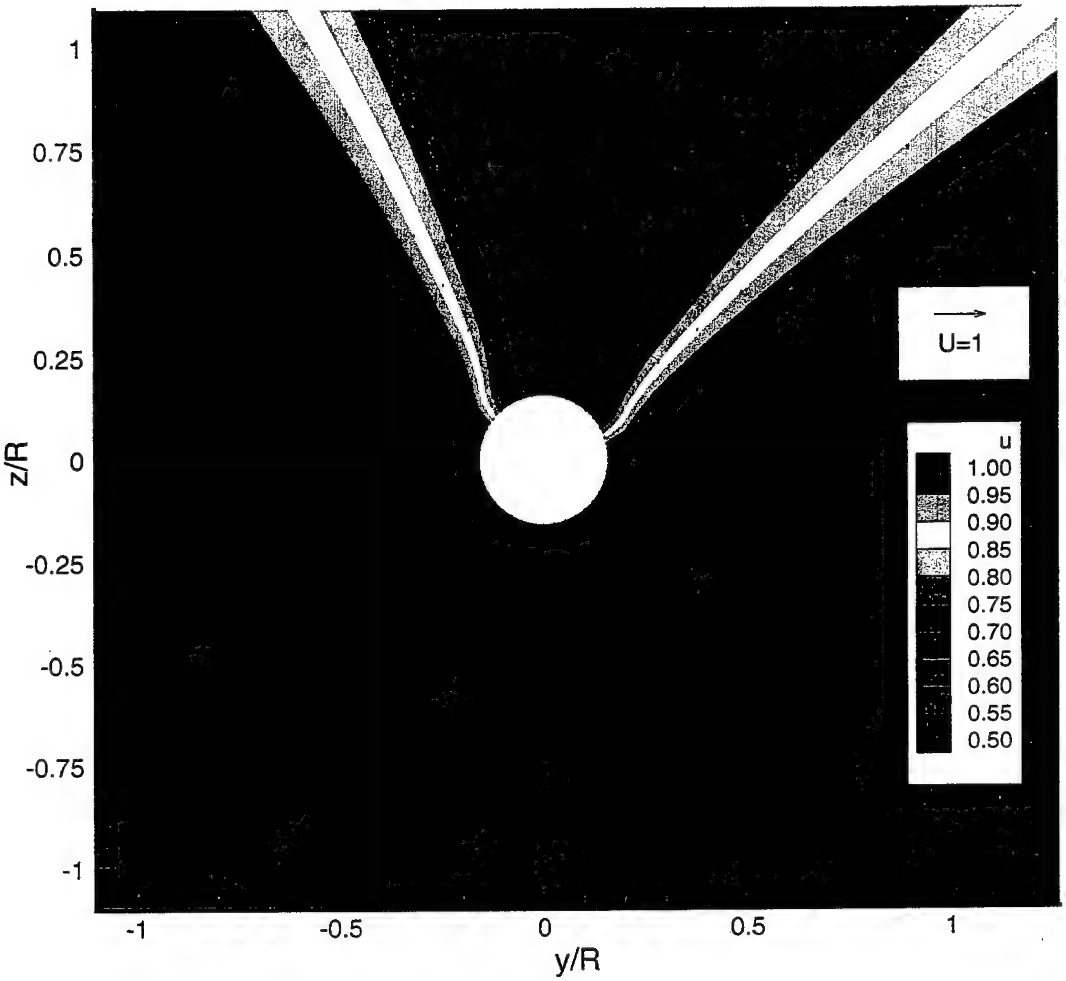

Fig. 7. "Wake" calculation, velocity field upstream of the propeller.

As the blade of the propeller encounters the wake, the effective angle of attack increases and one expects to see an accompanying increase in blade forces. Figure 8 shows K_T and K_Q coefficients as a function of angular blade position for this inflow condition. Note that both coefficients achieve maximum at about θ_{blade}=-265°, that is as the blade approaches the location of the wake's peak. For the K_T maximum to coincide exactly with the angular location of the wake center we would expect θ_{blade}=-280°. The apparent 15° mismatch is simply a consequence of the fact that the blade loading is largest near the leading edge while the blade angular location is defined with respect to the generator line which is closer to the mid-chord. Indeed, after the propeller forces are transferred to the RANS grid the body force density exhibits a maximum at the expected angular position matching the location of the maximum velocity deficit (see Figure 9). This experiment confirms that our code correctly transfers data between respective RANS and PUF coordinate systems. The increased force density caused by the blade's interaction with the wake is a source of an additional acceleration of the flow. This has a smoothing effect on the velocity field. In fact, as Figure 10 demonstrates, the maximum axial and radial velocity components downstream of the propeller seem to be the largest at the location of the inflow wake. This illustrates the important difference between the steady and unsteady RANS-propeller interaction codes. Since steady code is limited to the zeroth harmonic, all local effects such as wakes are averaged out. The unsteady approach is necessary if one wants to predict the time averaged location and velocity field of body wakes as they pass through the propeller.

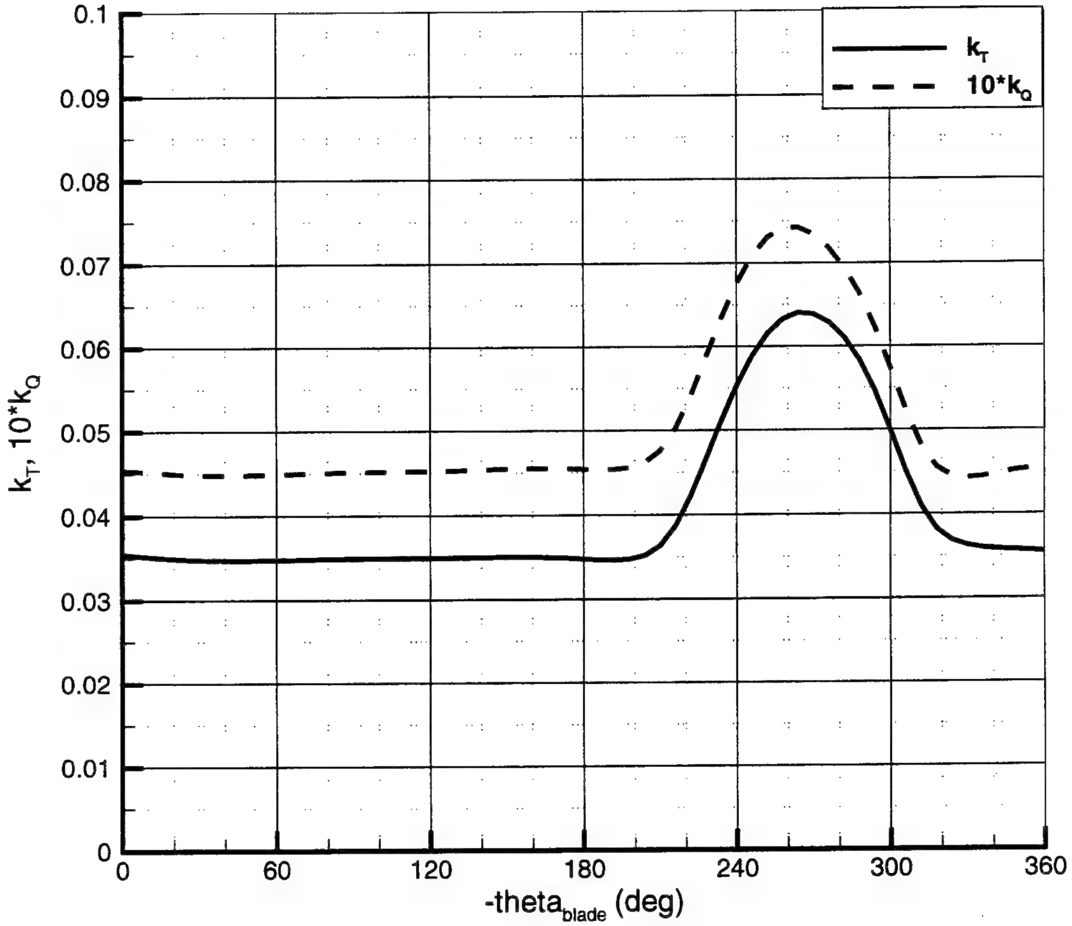

Fig. 8. "Wake" calculation, MARIN B-series propeller. Thrust and torque coefficients as a function of shaft angle.

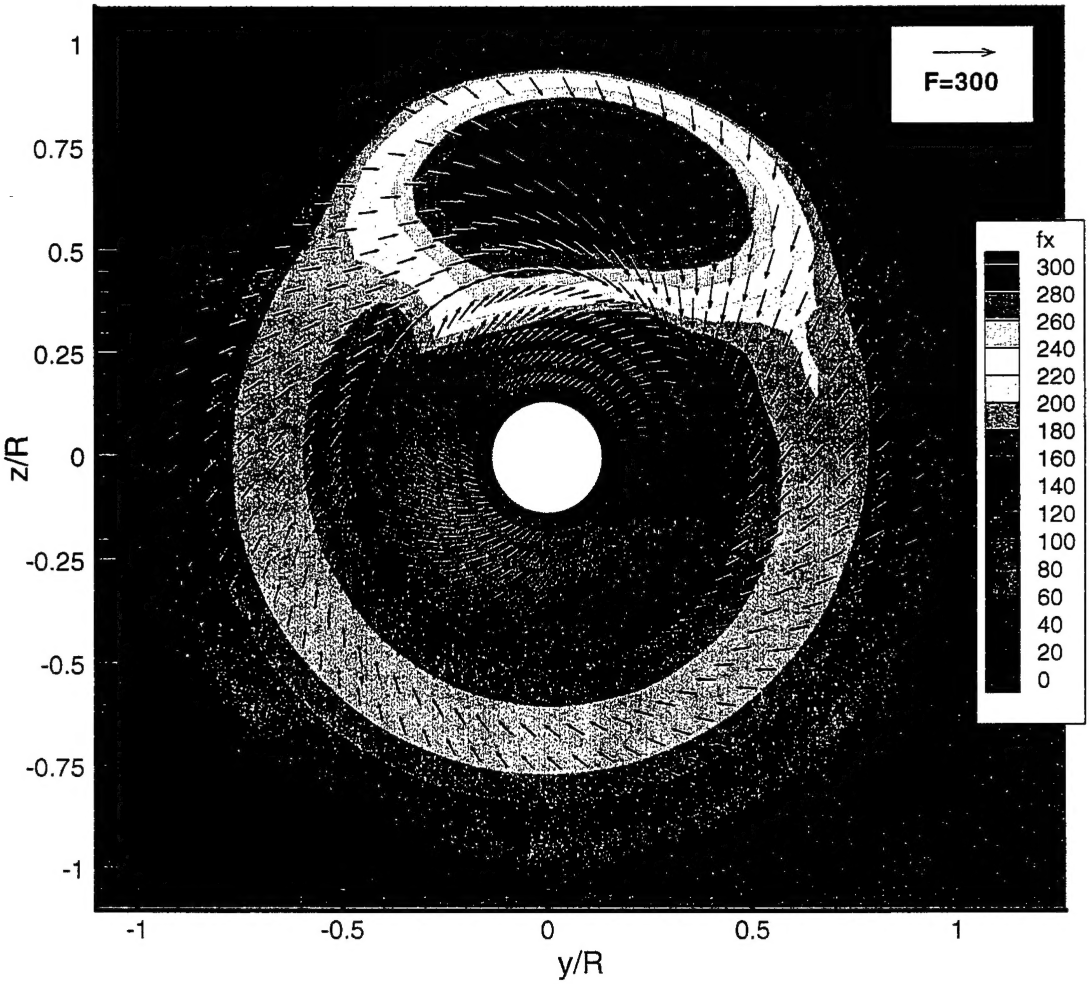

Fig. 9. "Wake" calculation, RANS body force distribution.

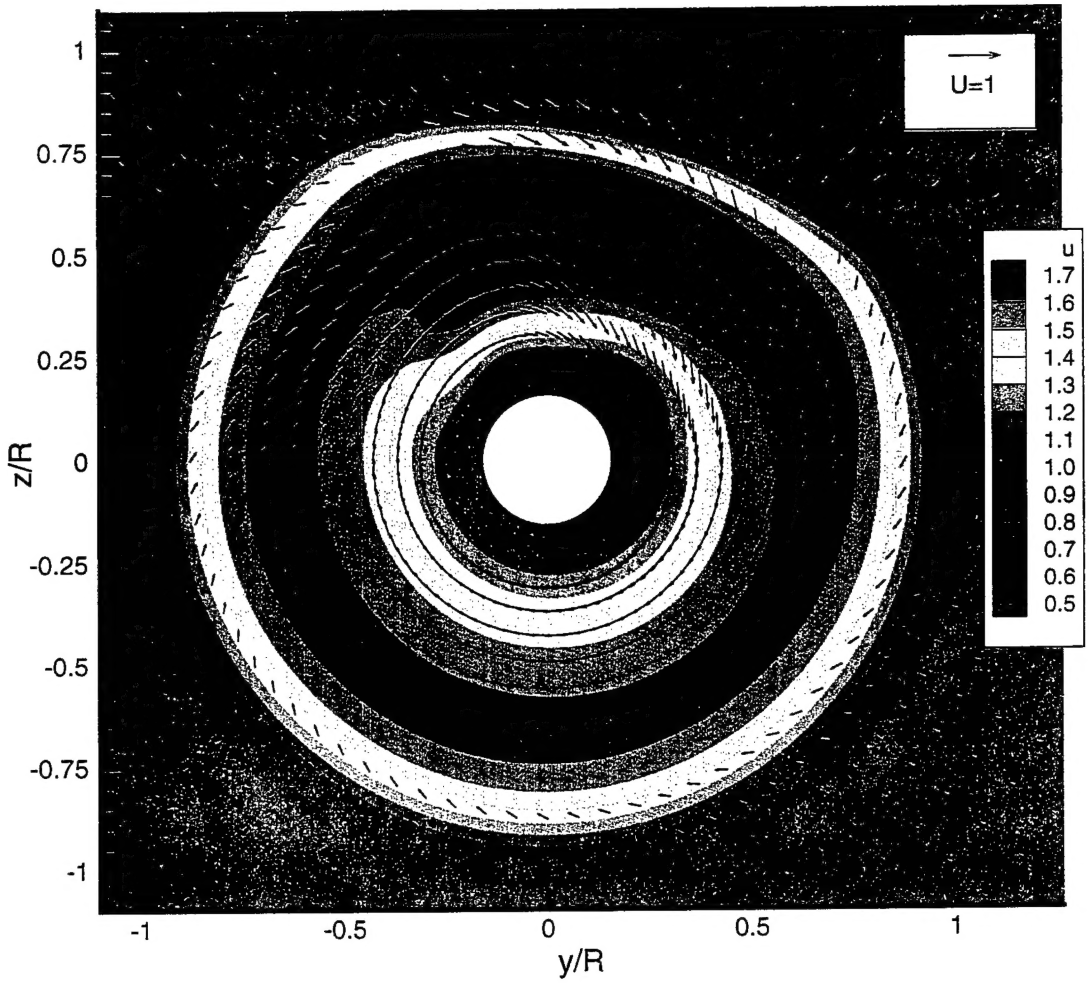

Fig. 10. "Wake" calculation, velocity field downstream of the propeller.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

The primary goal of this work, the coupling of a RANS flow solver with an unsteady propulsor force code, has been achieved. The RANS code needs body forces at the correct grid locations and the unsteady force code needs the RANS computed inflow. As shown, there are several steps between the two endpoints with the allowance of circumferentially varying inflow and body forces adding an extra degree of complexity. The present effort deals with this process and we feel that much of the code generated is modular enough to be used with other RANS and propulsor force codes besides the ones demonstrated here.

The preliminary numerical experiments presented here by no means comprise an exhaustive set of tests required for a complete evaluation of our code. Practical RANS grids typically require the flow domain to be divided into several separate zones with high density of grid points in selected regions of the flow to resolve large local gradients. Hence, grids having cell volumes varying by ten or more orders of magnitude are common. To ensure that our code properly handles such cases, a multi-zone, arbitrarily oriented grid with greatly varying cell sizes should be tested. The open water case can be rerun with the new grid to assure that the interpolation procedures provide sufficient precision and that they do not contain any hidden programming errors. If insufficient precision surfaces as a problem, the linear interpolation scheme used currently in the program should be replaced by a higher order method. In addition to checking the integrated quantities such as the thrust and torque coefficients, a detailed look at force distribution on the blade should be included as part of the exercise. As a final validation, a full propeller-body configuration should be gridded and calculated using our code and the results compared to the existing experimental data. The validated coupling code should then be extended to work with the next generation of propulsor and RANS software such as the new MIT propulsor code PUF-14 and the Mississippi State University RANS code UNCLE.

ACKNOWLEDGMENTS

The authors appreciate the encouragement of Professor Fred Stern at the University of Iowa.

REFERENCES

1. Stern, F., Kim, H. T., Patel, V. C., and Chen, H. C., “A Viscous-flow Approach to the Computation of Propeller-Hull Interaction”, Journal of Ship Research, Vol. 32, No. 4, pp. 246-262, (1988).
2. Tseng, C. L., Nguyen. P., and Fuhs, D., “Body forces and induced velocities for an FF1052 propeller”, Report CRDKNSWC/HD-0011-25, (May 1997).
3. Gorski, J. J, “Solutions of the incompressible Navier Stokes equations using an upwind differenced TVD scheme”, Lecture Notes in Physics, Vol. 323, pp. 278-282, (1988).
4. Keenan, D. P., “Propeller unsteady performance analysis program: MIT PUF2.1. Program documentation, MIT Report 81-1, (1987).
5. Weems, K. M., and Korpus, R. A., “A RANS Based Propeller/Hull Interaction Analysis System”, Science Applications International Corporation Report, SAIC-93/1068, (April 1993).
6. Carnahan, B., Luther, H. A., and Wilkes, J., “Applied Numerical Methods”, John Wiley & Sons, (1969).